Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 227 (2009) 179–193

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Reachability Analysis in BioAmbients

Gianluigi Zavattaro[1]

Dipartimento di Scienze dell'Informazione, Università di Bologna, Mura Anteo Zamboni 7, I-40127 Bologna, Italy

Abstract

We rephrase results, previously proved by Busi and Zavattaro on Mobile Ambients, characterizing a fragment of BioAmbients –without restriction and merge– in which a generalized form of reachability turns out to be decidable.

Keywords: reachability analysis, target reachability, decidability, ambients

1 Introduction

BioAmbients [11] is a well known formalism for the description of biological systems that combines the communication mechanisms of the π-calculus [10] with the notion of ambient as formalized in Mobile Ambients [7]. This combination permits to represent biochemical reactions by means of process communication and to model biological compartments by means of ambients. A bioambient $[P]$ is a collection of active processes and nested sub-bioambients P. Active processes can perform communication actions with other processes or execute capabilities in order to modify the ambient nesting. Communication consists of the interaction between an output and an input action performed by processes located in the same ambient, in parent/child ambients, or in two parallel ambients. The capabilities allow processes to modify ambient nesting in three possible ways: one ambient can move inside a parallel ambient, one ambient can move outside from the parent ambient, or two parallel ambients can merge into one single ambient.

In this paper we discuss the reachability problem in BioAmbients exploiting techniques developed by Busi and Zavattaro for Mobile Ambients [4,5] and already partially applied to small fragments of BioAmbients by Delzanno and Montagna [8].

[1] E-mail: zavattar@cs.unibo.it

doi:10.1016/j.entcs.2008.12.111

Reachability analysis consists in verifying, given two processes P and Q, whether there exists a computation that starts from P and leads to Q.

As an example of the kind of analysis that reachability supports, let us consider the following drug delivery scenario. A drug molecule, hosted into a specific transport molecule, is injected in the blood of a patient. The behavior of the hosting molecule should be such that the drug molecule is delivered to muscular tissue and not to connective tissue. Such a system could be represented by the following BioAmbient process

$$[[Drug] \mid Transport] \mid [Blood \mid [Muscular] \mid [Connective]]$$

where the first ambient represents the drug molecule hosted inside the transport molecule, while the second ambient represents (an abstraction of) the patient in which the blood is the transport medium to reach the muscular and the connective tissues.

We can formalize the expected behavior of the system saying that we expect that the configuration

$$[Blood' \mid [Transport']] \mid [Muscular' \mid [Drug]] \mid [Connective']]$$

should be reachable (where $Blood'$, $Transport'$, $Muscular'$, and $Connective'$ are some specific derivatives of the corresponding processes in the initial system configuration) while the following configuration should not be reachable

$$[Blood'' \mid [Transport'']] \mid [Muscular'']] \mid [Connective'' \mid [Drug]]]$$

(where $Blood''$, $Transport''$, $Muscular''$, and $Connective''$ are any possible derivatives of the corresponding processes in the initial system configuration).

Reachability is usually undecidable in Turing complete formalisms such as the π-calculus or Mobile Ambients. Nevertheless, at least for Mobile Ambients, very interesting fragments have been studied which are expressive enough to model all computable functions, but for which reachability turns out to be decidable. This holds for the fragment without restriction, without the *open* capability (used to dissolve an ambient boundary), and in which replication can be applied only to prefixed processes. In fact, this fragment was first proved to be Turing complete by Maffeis and Phillips [9], and then reachability was proved to be decidable for such fragment by Busi and Zavattaro [4]. This result follows from a monotonicity property of the considered calculus: because of the absence of the *open* capability, the number of "active" ambients in a process cannot decrease during the computation.

In a related paper [5] Busi and Zavattaro extended their previous result about the decidability of reachability for Mobile Ambients in two directions. On the one hand, they introduced a more general notion of reachability called *target reachability*. This allows for the specification of a possibly infinite class of targets specified by showing the nesting structure of ambients, and indicating lower and upper bounds to the number of specific instances of processes hosted in each ambient. On the other hand, the decidability of target reachability was proved for an extended calculus including also the sophisticated communication mechanisms of Boxed Ambients [2].

In this paper we rephrase the results proved in [5] applying them to BioAmbients. The main differences between the calculus considered in [5] and the fragment

of BioAmbients considered in this paper are: (i) monotonicity was obtained in [4,5] removing the *open* capability while it is obtained here for BioAmbients removing the *merge* capability; (ii) differently from Mobile Ambients (where a single process can execute its capability in order to move its hosting ambient), in BioAmbients two distinct processes located at different ambients must synchronize in order to allow their hosting ambients to change the nesting structure; (iii) communication in BioAmbients includes also communication between processes in sibling ambients while in Boxed Ambients communication is either local or parent-child; (iv) BioAmbients include also a choice operator (guarded on either communication actions or capabilities) while in the calculus considered in [5] there is no choice operator.

The proof of the decidability of target reachability is as in [5] by reduction to Petri net marking reachability (a decidable property for Petri nets). The reduction is obtained in two steps: we first define target marking reachability for Petri nets and we prove that it can be reduced to marking reachability, then we show how to reduce target reachability in BioAmbients to target marking reachability in Petri nets. Due to space limitation and because the proof is a small variant of the previous proof for Boxed Ambients, the details are omitted and can be found in [5] and in the full version of the paper [4] available at [6].

It is worth mentioning that this is not the first paper that applies the techniques developed by Busi and Zavattaro to BioAmbients. In fact, the technique developed in [4] has been already exploited by Delzanno and Montagna [8] to prove the decidability of (spatial) reachability for a fragment of BioAmbients without restriction, merge and without communication. Spatial reachability is a restricted version of target reachability in which only lower bounds to the number of the occurrences of processes in the target ambients can be specified. In this paper we prove a result which is more general than the one in [8] for three main reasons: we prove the decidability of target reachability which is more general than spatial reachability, we consider also the rather rich set of communication primitives of BioAmbients, and we consider the standard semantics instead of replacing the structural congruence rule for replication $!P \equiv !P|P$ with the reduction step $!P \rightarrow !P|P$ (the price we pay to maintain the standard semantics is to limit the syntax avoiding the application of replication to ambients, i.e. $![P]$ is not admitted in our calculus).

The paper is structured as follows. In Section 2 we report the syntax and semantics of BioAmb$^-$, the fragment of BioAmbients that we consider, and we recall the definition of target reachability. In Section 3 we prove that target reachability is decidable in BioAmb$^-$.

2 BioAmbients without Merge and Restriction

In this section we introduce a fragment of BioAmbients, called BioAmb$^-$, for which we prove the decidability of reachability.

Definition 2.1 – BioAmb$^-$ – Let $Name$, ranged over by n, m, p, ..., be a denumerable set of ambient names. The terms of BioAmbients are defined by the

following grammar:

π	::=			Actions
		$\$n!\{m\}$		Output action
		$\$n?\{m\}$		Input action
$\$$	::=			Directions
			local	Intra-ambients
			$s2s$	Inter-siblings
			$p2c$	Parent to child
			$c2p$	Child to parent
M, N	::=			Capabilities
			$enter\, n$	Synch entry
			$accept\, n$	Synch accept
			$exit\, n$	Synch exit
			$expel\, n$	Synch expel
P, Q	::=			Processes
		$P \mid Q$		Composition
		$[P]$		Ambient (membrane)
		G		Guarded process
		$!G$		Replication
G, L	::=			Guarded processes
		$\sum_{i \in I} \pi_i.P_i$		Communication choice
		$\sum_{i \in I} M_i.P_i$		Capability choice

We assume that in a process P the bound names, i.e., those names appearing as objects in input actions, are all pairwise distinct and disjoint from the free names. In this way, we can avoid to consider α-conversion.

We write $\pi.P$ for single communication choice, $\pi_1.P + \pi_2.Q$ for binary communication choice, $\mathbf{0}$ for empty choice, and $\pi.P + T$ to single out one communication option, and similarly for capability choice.

In the following we use $\prod_k P$ to denote the parallel composition of k instances of the process P, while $\prod_i P_i$ denotes the parallel composition of the indexed processes P_i.

BioAmbients processes run inside ambients and performs communication actions

and capabilities to modify the ambient structure. Communication is based on the standard input and output action à la π-calculus enriched with variants for allowing also processes running in different ambients to communicate. Namely, communication between parent-child and siblings is admitted. Capabilities are used to allow a process to move its hosting ambient outside (resp. inside) an outer (resp. a sibling) ambient. Namely, *exit* and *expel* are used for outside movement, while *enter* and *accept* are for inside movement. In the full BioAmbients also the *merge* capability is considered that permits two sibling ambients to merge their processes into a unique ambient.

Infinite behaviours in BioAmbients are modeled using the replication operator. In BioAmb$^-$ we do not admit the application of replication to ambients, e.g., $![P]$ is not a valid process. More precisely, the calculus BioAmb$^-$ corresponds to the fragment of BioAmbients [11] in which the restriction operator and the *merge* capability are not considered and in which replication can be applied only to guarded processes.

The operational semantics is defined in terms of a structural congruence plus a reduction relation.

Definition 2.2 – Structural congruence – The structural congruence $\equiv$ is the smallest congruence relation satisfying:

$$P \mid \mathbf{0} \equiv P \qquad P \mid Q \equiv Q \mid P$$

$$P \mid (Q \mid R) \equiv (P \mid Q) \mid R \qquad !P \equiv P \mid !P$$

Definition 2.3 – Reduction relation – The reduction relation is the smallest relation $\rightarrow$ satisfying the following axioms and rules:

$$[(T + enter\, n.P) \mid Q] \mid [(T' + accept\, n.R) \mid S] \rightarrow [[P \mid Q] \mid R \mid S]$$

$$[[(T + exit\, n.P) \mid Q] \mid (T' + expel\, n.R) \mid S] \rightarrow [P \mid Q] \mid [R \mid S]$$

$$(T + local\, n!\{m\}.P) \mid (local\, n?\{p\}.Q + T') \rightarrow P \mid Q\{m/p\}$$

$$(T + p2c\, n!\{m\}.P) \mid [(c2p\, n?\{p\}.Q + T') \mid R \rightarrow P \mid [Q\{m/p\} \mid R]$$

$$[R \mid (T + c2p\, n!\{m\}.P)] \mid (p2c\, n?\{p\}.Q + T') \rightarrow [R \mid P] \mid Q\{m/p\}$$

$$[R \mid (T + s2s\, n!\{m\}.P)] \mid [(s2s\, n?\{p\}.Q + T') \mid S] \rightarrow [R \mid P] \mid [Q\{m/p\} \mid S]$$

$$P \rightarrow Q \;\Rightarrow\; [P] \rightarrow [Q]$$

$$P \rightarrow Q \;\Rightarrow\; P|R \rightarrow Q|S$$

$$P \equiv P', P \rightarrow Q, Q \equiv Q' \;\Rightarrow\; P' \rightarrow Q'$$

where $Q\{m/p\}$ is the usual substitution of p for the free occurrences of m in Q.

The first two reduction rules handle ambient operations. The next four reduction rules handle communication within ambients (similar to the π-calculus) and between neighboring ambients. The remaining rules handle reductions in context and up to structural congruence.

In the following, we use $\rightarrow^*$ to denote the reflexive and transitive closure of $\rightarrow$. If $P \rightarrow^* Q$ we say that Q is a *derivative* of P.

2.1 Target Reachability

Classical reachability analysis consists in checking if $P \rightarrow^* R$ for two given processes P and R. In this paper we consider a more general notion of reachability. The main novelty is that we permit a partial description of the target process. More precisely, it is possible to impose constraints on the number of occurrences of guarded processes inside an ambient. Such constraints are both lower bounds (e.g. there must be at least one instance of the guarded process $M.P$ in a given ambient) and upper bounds (e.g. there can be at most two occurrences of the guarded process $\pi.Q$ in a given ambient).

We need to introduce some additional notation to denote the partial description of target processes.

We introduce a notion of normal form for processes that forbids the presence of both the unreplicated and the replicated version of a guarded term in a parallel composition. Any process can be transformed in a structurally congruent process in normal form by using the monoidal axioms for parallel composition and by applying the axiom for replication from right to left (i.e., $M.P \mid !M.P$ is transformed in $!M.P$).

Definition 2.4 – **Normal form** – A process P is in normal form if $P = \prod_i G_i \mid \prod_j !G'_j \mid \prod_k [P_k]$ and the following conditions hold:

- Q is in normal form for all Q such that $G_i = \pi.Q + R$ or $G_i = M.Q + R$ or $G'_j = \pi.Q + R$ or $G'_j = M.Q + R$ for some i, j, π, M, R;
- P_k is in normal form for all k;
- there exist no i, j such that $G_i = G'_j$.

Proposition 2.5 *Let P be a process. Then there exists a process Q in normal form such that $P \equiv Q$.*

Definition 2.6 – **Target** – The set of *targets* is defined by the following grammar:

$$T \ ::= \ \mathbf{0} \mid \texttt{any} \mid q \leq G \leq q' \mid !G \mid T|T \mid [T]$$

where $q \in I\!N$ and $q' \in I\!N \cup \{\infty\}$. [2]

A target `any` requires the presence of zero or more occurrences of any process, while $q \leq G \leq q'$ requires the presence of k occurrences of the guarded process G, with $q \leq k \leq q'$ (if $q' = \infty$ there is no upper bound to the number of occurrences). A target $!G$ requires the presence of one or more occurrences of process $!G$. As the behaviour of processes $\prod_k !G$ is the same for any $k \geq 1$, we prefer to require just the presence – or the absence – of a replicated process instead of providing upper

[2] $I\!N$ denotes the set of natural numbers and we assume that $q \leq \infty$ for all $q \in I\!N$.

and lower bounds to the number of its occurrences. Targets can be composed in parallel, and can be nested in ambients.

As an example, consider the target $[1 \leq \mathit{expel}\, n.P \leq 2 \mid [!G] \mid [\mathtt{any} \mid 3 \leq \mathit{exit}\, n.Q \leq \infty]]$. This target requires that an outer ambient contains one or two occurrences of process $\mathit{expel}\, n.P$, an ambient containing only occurrences of process $!G$ (at least one occurrence is required), and an ambient containing at least three occurrences of the process $\mathit{exit}\, n.Q$ and any other process. Moreover, this target also requires that there is no process at top level.

We consider only a proper subset of *well formed* targets defined as follows.

Basically, a target is well formed if the upper and lower bounds on guarded terms are satisfiable (i.e., target $3 \leq \pi.P \leq 2$ is not well formed) and if the presence of a replicated version of a guarded process prevents the occurrence of the nonreplicated version of the same process in a parallel composition (i.e., target $\pi.P \mid !\pi.P$ is not well formed). We also require that at most one occurrence of a replicated process is present in a parallel composition (i.e., target $!\pi.P \mid !\pi.P$ is not well formed).

Definition 2.7 – Well formed target – A *target* T is well formed if there exists a target $S = \prod_i q_i \leq G_i \leq q'_i \mid \prod_j !G'_j \mid \prod_k [T'_k]$ such that the following conditions hold:

- processes G_i, G'_j are in normal form for all i, j;
- either $T \equiv S$ or $T \equiv S \mid \mathtt{any}$;
- $q_i \leq q'_i$ for all i;
- there exist no i, j such that $G_i = G'_j$;
- if $G'_j = G'_{j'}$ then $j = j'$;
- T'_k is well formed for all k.

We define the set of processes $\mathit{set}(T)$ that satisfy the constraints imposed by a target T. Basically, we require the presence of the required number of occurrences of a guarded process in each ambient; if the upper bound is ∞, then also the presence of a replicated version of the process satisfies the target (i.e., process $[!G]$ satisfies the target $[3 \leq G \leq \infty]$). If the target `any` is present, then further (different) processes may be present. As already discussed, with a replicated process in the target we just require the presence of at least one occurrence of such a replicated process.

Definition 2.8 – set(T) – Let T be a well formed target. A process P is in $\mathit{set}(T)$ if $P \equiv \prod_h L_h \mid \prod_g !L'_g \mid \prod_k [P'_k]$ and there exists a target $S = \prod_i q_i \leq G_i \leq q'_i \mid \prod_j !G'_j \mid \prod_k [T'_k]$ such that the following conditions hold:

- either $T \equiv S$ or $T \equiv S \mid \mathtt{any}$;
- for all i, either $q_i \leq |\{h \mid L_h = G_i\}| \leq q'_i$ or $q'_i = \infty$ and there exists g such that $L'_g = G_i$;
- for all j there exist g such that $L'_g = G'_j$;
- if $T \equiv S$ then for any h there exists i such that either $L_h = G_i$ or $L_h = G'_i$

and for any g there exists j such that $L'_g = G'_j$;

- for any k, $P'_k \in set(T'_k)$.

It is worth to note that $set(T)$ is compatible with the structural congruence relation as formalized by the following Proposition.

Proposition 2.9 *Let T be a well formed target and P and Q two processes such that $P \equiv Q$. Then, $P \in set(T)$ if and only if $Q \in set(T)$.*

We are now ready to formalize the notion of *target reachability*.

Definition 2.10 Let P be a process and T be a well formed target. We say that T is a target reachable from P (denoted by $TReach(P, T)$) if there exists a process Q such that $P \rightarrow^* Q$ and $Q \in set(T)$.

3 Deciding target reachability in BioAmb$^-$

The target reachability problem for BioAmb$^-$ processes consists in checking if, given a target T and a process P, the target T is reachable from P. In this Section we show that target reachability is decidable for BioAmb$^-$ processes. The proof is basically an adaptation of the proof of decidability of reachability for a fragment of Boxed Ambients considered in [5]. The main differences are due the presence of different kinds of communication mechanisms (e.g. sibling-to-sibling) and different capabilities (e.g. the pair *expel*, *exit* instead of the *out* capability). This decidability result is proved showing how to reduce target reachability on BioAmb$^-$ to marking reachability on Petri nets. This is obtained in two steps, we first recall the notion of *target marking reachability* for Petri nets defined in [5], then we reduce target reachability in BioAmb$^-$ into target marking reachability into Petri nets.

We start recalling some basic definitions on Petri nets, then we define target marking reachability and we provide a sketch of the reduction result.

3.1 P/T Nets

We recall Place/Transition nets with unweighted flow arcs (see, e.g., [12]). Here we provide a characterization of this model which is convenient for our aims.

Definition 3.1 Given a set S, a *finite multiset* over S is a function $m : S \rightarrow I\!N$ such that the set $dom(m) = \{s \in S \mid m(s) \neq 0\}$ is finite. The *multiplicity* of an element s in m is given by the natural number $m(s)$. The set of all finite multisets over S, denoted by $\mathcal{M}_{fin}(S)$, is ranged over by m. A multiset m such that $dom(m) = \emptyset$ is called *empty*. The set of all finite sets over S is denoted by $\wp_{fin}(S)$.

Given the multiset m and m', we write $m \subseteq m'$ if $m(s) \leq m'(s)$ for all $s \in S$ while $\oplus$ denotes their *multiset union*: $m \oplus m'(s) = m(s) + m'(s)$. The operator $\setminus$ denotes *multiset difference*: $(m \setminus m')(s) =$ if $m(s) \geq m'(s)$ then $m(s) - m'(s)$ else 0. The *scalar product*, $j \cdot m$, of a number j with m is $(j \cdot m)(s) = j \cdot (m(s))$.

To lighten the notation, we sometimes use the following abbreviation. If m is a multiset containing only one occurrence of an element s (i.e., $dom(m) = \{s\}$ and

$m(s) = 1$) we denote m by only s. Multiset union is represented also by comma, i.e., $m, m' = m \oplus m'$. Let m be a multiset over S and m' a multiset over $S' \supseteq S$, such that $(m'(s') = 0)$ for each $s' \in S' \setminus S$; with abuse of notation, we sometimes use m in place of m', and vice versa.

Definition 3.2 A P/T net is a pair (S, T) where S is the set of *places* and $T \subseteq \mathcal{M}_{fin}(S) \times \mathcal{M}_{fin}(S)$ is the set of *transitions*.

Finite multisets over the set S of places are called *markings*. Given a marking m and a place s, we say that the place s contains $m(s)$ *tokens*.

A P/T net is finite if both S and T are finite.

A P/T system is a triple $N = (S, T, m_0)$ where (S, T) is a P/T net and m_0 is the *initial marking*.

A transition $t = (c, p)$ is usually written in the form $c \to p$. The marking c, usually denoted by ${}^\bullet t$, is called the *preset* of t and represents the tokens to be *consumed*; the marking p, usually denoted by $t^\bullet$, is called the *postset* of t and represents the tokens to be *produced*.

A transition t is *enabled* at m if ${}^\bullet t \subseteq m$. The execution of a transition t enabled at m produces the marking $m' = (m \setminus {}^\bullet t) \oplus t^\bullet$. This is written as $m \xrightarrow{t} m'$ or simply $m \to m'$ when the transition t is not relevant. We use σ, τ to range over sequences of transitions; the empty sequence is denoted by ε; let $\sigma = t_1, \ldots, t_n$, we write $m \xrightarrow{\sigma} m'$ to mean the *firing sequence* $m \xrightarrow{t_1} \cdots \xrightarrow{t_n} m'$.

We say that m' is *reachable from* m if there exists σ such that $m \xrightarrow{\sigma} m'$.

We say that m' *covers* m if $m \subseteq m'$.

Definition 3.3 Let $N = (S, T, m_0)$ be a P/T system.

The reachability problem for marking m consists of checking if $m_0 \to^* m$.

The coverability problem for marking m consists of checking if there exists m' such that $m_0 \to^* m'$ and m' covers m.

3.2 Target marking reachability on P/T nets

We introduce a generalization of both the notions of reachability and coverability on P/T nets. The idea essentially consists in providing a lower and an upper bound to the number of tokens in each place of the net, and in checking if it is possible to reach a marking that satisfies such constraints.

Definition 3.4 – target marking – Let $N = (S, T)$ be a P/T net. A target marking of N is a pair of functions $(inf, sup) \in (S \to \mathbb{N}) \times (S \to \mathbb{N} \cup \infty)$ such that, for all $s \in S$, $inf(s) \leq sup(s)$.

Definition 3.5 – target marking satisfiablity – Let $N = (S, T)$ be a P/T net. A marking m of N satisfies a target marking (inf, sup) of N if, for all $s \in S$, $inf(s) \leq m(s) \leq sup(s)$.

Definition 3.6 – target marking reachability – Let $N = (S, T, m_0)$ be a P/T system. A target marking (inf, sup) is reachable if there exists a marking m such that $m_0 \to^* m$ and m satisfies (inf, sup).

We note that reachability and coverability are special cases of target marking reachability. Checking reachability of marking m is equivalent to check reachability of the target marking (m, m), while checking coverability of m is equivalent to reachability of the target marking $(m, \{(s, \infty) \mid s \in S\})$.

Now we reduce the target marking reachability problem for a system N and a target marking (inf, sup) to standard reachability on the P/T system $TMSys(N, (inf, sup))$ defined below.

Definition 3.7 Let $N = (S, T, m_0)$ be a P/T system and (inf, sup) be a target marking of N. The P/T system $TMSys(N, (inf, sup)) = (S', T', m'_0)$ is defined as follows. Let $normal, ending \notin S$.

$$\begin{aligned} S' = \;& S \cup \{normal, ending\} \\ T' = \;& \{(c \cup normal, p \cup normal) \mid (c, p) \in T\} \cup \\ & \{(normal, ending)\} \cup \\ & \{(s \cup ending, ending) \mid sup(s) = \infty\} \\ m'_0 = \;& m_0 \cup normal \end{aligned}$$

The set of markings $TMMark(N, (inf, sup))$ is defined as follows:

$$\begin{aligned} TMMark(N, (inf, sup)) = \{m \mid \forall s \in S : (sup(s) = \infty \Rightarrow m(s) = inf(s)) \wedge \\ (sup(s) \neq \infty \Rightarrow inf(s) \leq m(s) \leq sup(s))\} \end{aligned}$$

Proposition 3.8 *Let $N = (S, T, m_0)$ be a P/T system and (inf, sup) be a target marking of N. The set of markings $TMMark(N, (inf, sup))$ is finite.*

Proposition 3.9 *Let $N = (S, T, m_0)$ be a P/T system and (inf, sup) be a target marking of N. The target marking (inf, sup) is reachable in N iff one of the markings in the set $TMMark(N, (inf, sup))$ is reachable in $TMSys(N, (inf, sup))$.*

As a consequence of the two propositions above and of the decidability of reachability on P/T systems, we get the following:

Corollary 3.10 *Target marking reachability is decidable for P/T systems.*

3.3 Reducing target reachability on processes to target marking reachability on P/T nets

Now we show that target reachability on processes can be reduced to target marking reachability on Petri nets; by decidability of target marking reachability on Petri nets, we get the following:

Theorem 3.11 *Let P be a BioAmb$^-$ process and T be a target. The target reachability problem $TReach(P, T)$ is decidable.*

Given a process P and a target R, we outline the construction of a (finite) Petri system $Sys_{P,R}$ satisfying the following property: the check of $TReach(P, T)$ is

equivalent to check target marking reachability of a (finite set of) target markings on $Sys_{P,R}$. The technical details concerning the construction of the net are quite similar to the ones for deciding reachability in the fragment of Mobile Ambients and Boxed Ambients considered in [4,6,5], and thus omitted. Here we only recall the basic ideas.

The intuition behind this result relies on the monotonicity of BioAmb$^-$: because of the absence of the *merge* capability, the number of "active" ambients in a process (i.e., ambients that are not guarded by any capability or communication) cannot decrease during the computation. Moreover, as the applicability of replication is restricted to guarded processes, the number of "active" ambients in a set of structurally equivalent processes is finite (while this is not the case in, e.g., the process $![G]$). Thanks to the property explained above, in order to check target reachability it is sufficient to take into account a subset of the derivatives of P: namely, the P-derivatives whose number of active ambients is not greater than the number of active ambients in the target.

Unfortunately, this subset of P-derivatives is, in general, not finite, as the processes inside an ambient can grow unlimitedly. Consider, e.g., the process $P = [!local\, n!\{m\} \mid !local\, n?\{p\}.Q]$: it is easy to see that, for any k, $[!local\, n!\{m\} \mid !local\, n?\{p\}.Q \mid \prod_k Q\{m/p\}]$ is a derivative of P.

On the other hand, we note that the set of guarded and replicated terms that can occur as subprocesses of (the derivatives of) a process P (namely, the subterms of kind G or $!G$) is finite. The idea is to borrow a technique used to map (the fragment without restriction of) a process algebra on Petri nets. A process P is decomposed in the (finite) multiset of its guarded and replicated subprocesses that appear at top-level (i.e., occur unguarded in P); this multiset is then considered as the marking of a Place/Transition Petri net. The execution of a computational step in a process will correspond to the firing (execution) of a transition in the corresponding net. Thus, we reduce the target reachability problem for BioAmb$^-$ processes to reachability of a finite set of target markings in a Place/Transition Petri net, which we have shown to be a decidable problem. However, differently from what happens in process algebras, where processes can be faithfully represented by a multiset of subprocesses, BioAmb$^-$ processes have a tree-like structure that hardly fits in a flat model such as a multiset.

The solution is to consider BioAmb$^-$ processes as composed of two kinds of components; the tree-like structure of ambients and the family of multisets of guarded and replicated subterms contained at top level in each ambient. As an example, consider the process

$$p2c\, n!\{m\}.P \mid [enter\, k.Q \mid G] \mid [accept\, k.\mathbf{0}] \mid [c2p\, n?\{p\}.R \mid [\mathbf{0}]]$$

having the tree-like structure $[] \mid [] \mid [[]]$. Moreover, there is a multiset corresponding to each "node" of the tree: the multiset $\{p2c\, n!\{m\}.P\}$ is associated to the root, $\{enter\, k.Q, G\}$ is associated to the first son of the root, $\{accept\, k.\mathbf{0}\}$ is associated to the second son of the root, $\{c2p\, n?\{p\}.R\}$ is associated to the third son of the root, and the empty multiset $\{\}$ to the son of the third son of the root.

The Petri net we construct is composed of the following two parts: the first part is basically a finite state automaton, where the marked place represents the current tree-like structure of the process; the second part is a set of identical subnets: the marking of each subnet represents the multiset associated to a particular node of the tree. To keep the correspondence between the nodes of the tree and the multiset associated to that node, we make use of labels. A distinct label is associated to each subnet; this label will be used in the tree-like structure to label the node whose contents (i.e., the set of guarded and replicated subprocesses contained in the ambient corresponding to the node) is represented by the subnet.

The set of possible tree-like structures we need to consider is finite, because to verify target reachability we need to take into account only those processes whose number of active ambients is limited by the number of active ambients in the target. The upper bound on the number of nodes in the tree-like structures also provides an upper bound to the number of identical subnets (at most one for each active ambient). In general, the number of active ambients grows during the computation; hence, we need a mechanism to remember which subnets are currently in use and which ones are not used. When a new ambient is created, a correspondence between the node representing such a new ambient in the tree-like structure and a not yet used subnet is established, and the places of the "fresh" subnet are filled with the marking corresponding to the guarded and replicated subprocesses contained in the newly created ambient. To this aim, each subnet is equipped with a place called *unused*, that contains a token as long as the subnet does not correspond to any node in the tree-like structure.

For example, consider the process $[accept\, n] \mid [enter\, n.[!expel\, k]]$. The relevant part of the net is depicted in Figure 1: a subset of the places, representing the tree-like structure, is depicted in the left-hand part of the figure, while the subnets are depicted in the right-hand part. We only report the subnets labelled with l_1, l_2, and l_3, and omit the subnet labelled with l_0 with empty marking. The computation step $[accept\, n] \mid [enter\, n.[!expel\, k]] \rightarrow [[[!expel\, k]]]$ corresponds to the firing of the transition enabled in the depicted net.

Now we are ready to describe the net that will be used to decide reachability of a target T starting from a process P.

The set of places of the net is constructed as follows. The part of the net representing the tree-like structure contains a place for each tree of size not greater than the number of active ambients in T. Each of the subnets contains a place for each guarded and replicated subprocess of process P, and a place named "unused", that remains filled as long as the subnet does not correspond to any node in the tree-like structure. Moreover, we associate a distinct label to each subnet, and all the places of the subnet will be decorated with such a label.

The net has three sets of transitions: the first set permits to model the execution of the capabilities, the second set is used deal with communication, and the third set to cope with replication.

We concentrate on the first set of transitions. A pair of capabilities, say, $enter\, n$ and $accept\, n$, can be executed when the following conditions are fulfilled: the tree-

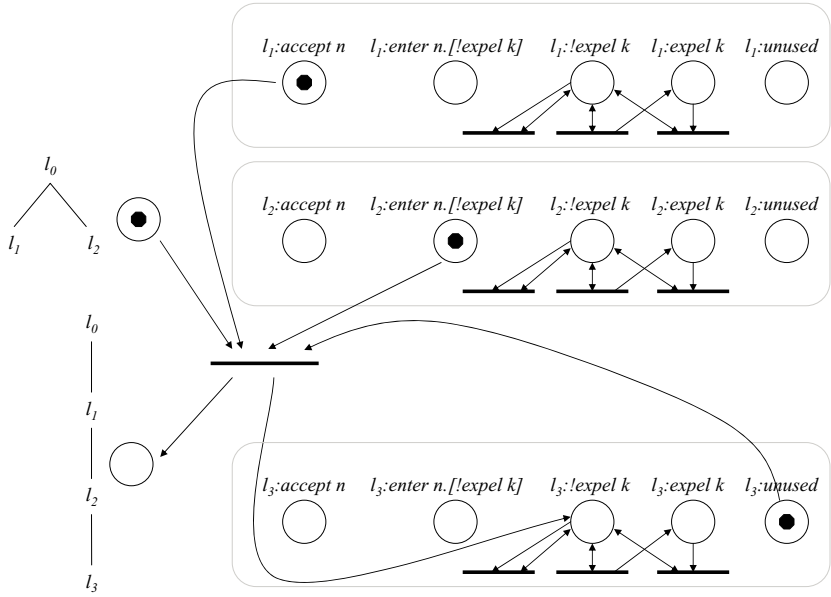

Fig. 1. A portion of the net corresponding to process $[accept\, n] \mid [enter\, n.[!expel\, k]]$.

like structure must have a specific structure with siblings l_i and l_j, a place corresponding to a guarded subprocess $enter\, n.P + T$ is marked in the subnet labeled l_i, and a place $accept\, n.Q + T'$ is marked in the subnet labeled l_j. Moreover, the number of active ambients created by the execution of the capability, added to the number of currently active ambients, must not exceed the number of active ambients in the target T. This condition is checked by requiring that there exist a sufficient number of "unused" places that are currently marked. The execution of the capability causes the following changes to the marking of the net: the place corresponding to the new tree-like structure is now filled and the marking of the subnets performing the $enter\, n$ and $accept\, n$ operations are updated (by adding the tokens in the places corresponding to the active guarded and replicated subprocesses in the continuations P and Q). Moreover, a number of subnets equal to the number of active ambients in the continuations P and Q become active: their places will be filled with the tokens corresponding to the active guarded and replicated subprocesses contained in the corresponding ambient, and the tree-like structure is updated accordingly.

The second set of transitions deal with communication and are quite similar to the rules for capabilities. A local communication can be executed when two places, corresponding to the guarded processes $local\, n!\{m\}.Q_1$ and $local\, n!\{p\}.Q_2$, are marked in a subnet, and the number of active ambients created by the execution of the communication, added to the number of currently active ambients, must not exceed the number of active ambients in the target T. The execution of the local

communication causes the following changes to the marking of the net: the place corresponding to the new tree-like structure is now filled and the marking of the subnet performing the communication is updated (by adding the tokens in the places corresponding to the active guarded and replicated subprocesses in the continuations Q_1 and Q_2). Moreover, a number of subnets equal to the number of active ambients in the continuations Q_1 and Q_2 become active. Nonlocal communication is more involved, because the two communicating processes reside in two different ambients and it is necessary to check that the two hosting ambients are located in the right places in the tree like structure.

The third set of transitions deals with replication. For all replicated processes $!G$ occurring in P, we add to each subnet the transitions $!G \rightarrow !G, G$ and $!G, G \rightarrow !G$ and $!G, !G \rightarrow !G$, respectively permitting to spawn a new copy of a replicated process, to absorb a process that also appears in a replicated form in the marking, and to remove multiple occurrences of a replicated process in a marking. These transitions are used to reduce target reachability on BioAmb$^-$ to target marking reachability on the net system. An instance of such transitions is depicted in the subnets of Figure 1.

The reachability of target T is reduced to reachability of a target marking (inf_T, sup_T) constructed as follows. We require that a token is contained in a place corresponding to the tree-like structure of T (and that the places corresponding to the other tree-like structures are empty). Moreover, for any active ambient in T,

- for any target $q \leq G \leq q'$ at top level in the active ambient, we require that $inf_T(l : G) = q$ and $sup_T(l : G) = q'$, where l is the label of the subnet corresponding to the active ambient;
- for any target $!G$ at top level in the active ambient, we require that $inf_T(l :!G) = sup_T(l :!G) = 1$;
- for any guarded or replicated process Q not occurring at top level in the active ambient, we require that $inf_T(l : Q) = 0$; if the target `any` occurs at top level in the active ambient, then we require $sup_T(l : Q) = \infty$, otherwise we impose $sup_T(l : Q) = 0$.

Acknowledgements

Research partially supported by the University of Bologna Strategic Project *CompReNDe*: Compositional and Executable Representations of Nano Devices.

This paper was written in memory of Nadia Busi who passed away the 5th of September 2007, at the age of 39. This paper is an example of how her work in the area of the theory of concurrency represents a precious source of inspiration for proving new interesting results also in the more recent area of biologically inspired process calculi. These calculi attracted the interest of Nadia in her very active last few years of scientific activity: besides writing many valuable papers on the expressiveness of such calculi, she has been also the organizer of the first edition of the meeting *MeCBIC* dedicated to *Membrane Computing and Biologically Inspired*

Process Calculi.

References

[1] I. Boneva and J.-M. Talbot. When Ambients Cannot be Opened. *Theoretical Computer Science*, 333(1-2):127–169, Elsevier, 2005.

[2] M. Bugliesi, G. Castagna and S. Crafa Access Control for Mobile Agents: The Calculus of Boxed Ambients. ACM Transactions on Programming Languages and Systems, 26(1):57-124. ACM Press, 2004.

[3] N. Busi and G. Zavattaro. On the Expressive Power of Movement and Restriction in Pure Mobile Ambients. *Theoretical Computer Science*, 322:477–515, Elsevier, 2004.

[4] N. Busi and G. Zavattaro. Deciding Reachability in Mobile Ambients. In *Proc. ESOP'05*, volume 3444 of *Lecture Notes in Computer Science*, pages 248-262. Springer-Verlag, Berlin, 2005.

[5] N. Busi and G. Zavattaro. Reachability Analysis in Boxed Ambients. In *Proc. ICTCS'05*, volume 3701 of *Lecture Notes in Computer Science*, pages 143–159. Springer-Verlag, Berlin, 2005.

[6] N. Busi and G. Zavattaro, Deciding Reachability in Mobile Ambients - Extended version. Available at http://www.cs.unibo.it/~busi/papers/MA05.pdf

[7] L. Cardelli and A.D. Gordon. Mobile Ambients. *Theoretical Computer Science*, 240(1):177–213, 2000.

[8] G. Delzanno and R. Montagna. On Reachability and Spatial Reachability in Fragments of BioAmbients, In *Proc. MeCBIC'06*, volume 171(2) of *Electronic Notes in Theoretical Computer Science*, pages 69–79. Elsevier, 2006.

[9] S. Maffeis and I. Phillips. On the Computational Strength of Pure Ambient Calculi. *Theoretical Computer Science*, 330(3):501-551, Elsevier, 2005.

[10] R. Milner, J. Parrow, D. Walker. A calculus of mobile processes. *Journal of Information and Computation*, 100:1–77. Academic Press, 1992.

[11] A. Regev, E.M. Panina, W. Silverman, L. Cardelli, and E.Y. Shapiro. BioAmbients: an abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141–167, Elsevier, 2004.

[12] W. Reisig. *Petri nets: An Introduction*. EATCS Monographs in Computer Science, Springer, 1985.